

<https://helda.helsinki.fi>

An Answer Set Programming Approach to Argumentative Reasoning in the ASPIC+ Framework

Lehtonen, Tuomo

International Joint Conferences on Artificial Intelligence
2020

Lehtonen , T , Wallner , J P & Järvisalo , M 2020 , An Answer Set Programming Approach to Argumentative Reasoning in the ASPIC+ Framework . in D Calvanese , E Erdem & M Thielscher (eds) , Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning (KR 2020) . International Joint Conferences on Artificial Intelligence , pp. 636-646 , International Conference on Principles of Knowledge Representation and Reasoning , Rhodes , Greece , 12/09/2020 . <https://doi.org/10.24963/kr.2020/63>

<http://hdl.handle.net/10138/326066>
<https://doi.org/10.24963/kr.2020/63>

unspecified
publishedVersion

Downloaded from Helda, University of Helsinki institutional repository.

This is an electronic reprint of the original article.

This reprint may differ from the original in pagination and typographic detail.

Please cite the original version.

An Answer Set Programming Approach to Argumentative Reasoning in the ASPIC⁺ Framework

Tuomo Lehtonen¹, Johannes P. Wallner², Matti Järvisalo¹

¹University of Helsinki, Finland

²TU Wien, Austria

{tuomo.lehtonen, matti.jarvisalo}@helsinki.fi, wallner@dbai.tuwien.ac.at

Abstract

A major research direction in AI argumentation is the study and development of practical computational techniques for reasoning in different argumentation formalisms. Compared to abstract argumentation, developing algorithmic techniques for different structured argumentation formalisms, such as assumption-based argumentation and the general ASPIC⁺ framework, is more challenging. At present, there is a lack of efficient approaches to reasoning in ASPIC⁺. We develop a direct declarative approach based on answer set programming (ASP) to reasoning in an instantiation of the ASPIC⁺ framework. We establish formal foundations for direct declarative encodings for reasoning in ASPIC⁺ without preferences for several central argumentation semantics, and detail ASP encodings of semantics for which reasoning about acceptance is NP-hard in ASPIC⁺. Empirically, the ASP approach scales up to frameworks of significant size, thereby answering the current lack of practical computational approaches to reasoning in ASPIC⁺ and providing a promising base for capturing further generalizations within ASPIC⁺.

1 Introduction

AI argumentation is today a significant area of knowledge representation and reasoning research (Baroni et al. 2018; Atkinson et al. 2017). A major research direction in AI argumentation is the study and development of practical computational techniques for reasoning in different argumentation formalisms. In terms of formalisms, significant research advances in the development of practical reasoning techniques have been achieved in recent years in particular in the context of Dung’s theory of abstract argumentation frameworks (AFs) (Dung 1995). This progress is witnessed by, e.g., various efficient system implementations and a biannual system competition focusing on central reasoning tasks such as credulous and skeptical acceptance of arguments (Thimm and Villata 2017; Gaggl et al. 2020; Cerutti et al. 2018). However, while AFs have proven to be very influential as a formal approach to argumentation, the level of abstraction inherent to AFs does not directly support specifying how arguments and attacks between arguments are constructed from knowledge bases. In contrast, various formalisms developed for structured argumentation—including ASPIC⁺ (Modgil and Prakken 2018; Modgil and Prakken 2013), assumption-based argumentation (ABA) (Cyras et al. 2018; Bondarenko et al.

1997), defeasible logic programming (DeLP) (García and Simari 2018; García and Simari 2004), and deductive argumentation (Besnard and Hunter 2018; Besnard and Hunter 2008)—aim at a more comprehensive approach with the ability to capture derivation of arguments and attacks starting from knowledge bases. However, from the computational perspective, this makes the development of practical reasoning approaches a more daunting task. Indeed, while algorithmic approaches for different variants of structured formalisms have been developed (Cerutti et al. 2018), it can be argued that practical systems for reasoning over structured formalisms are currently at a less mature state than those developed for abstract argumentation.

Motivated by advances in employing fully declarative techniques for reasoning in AFs (Charwat et al. 2015), a first fully declarative approach—based on answer set programming (ASP) techniques (Niemelä 1999; Gelfond and Lifschitz 1988; Brewka et al. 2015)—to reasoning in the structured formalism of ABA was recently developed (Lehtonen, Wallner, and Järvisalo 2019). This ASP-based approach to ABA was shown to significantly improve on the earlier state-of-the-art systems for ABA reasoning in terms of empirical performance. This motivates further study of the applicability of ASP to capture reasoning in structured argumentation formalisms more generally.

With these motivations, the focus of this paper is on developing an ASP-based approach to reasoning in the rule-based structured argumentation formalism of ASPIC⁺. The ASPIC⁺ framework finds application avenues, e.g., in legal reasoning (Prakken et al. 2015; Prakken 2012), ontology-based data access (Yun and Croitoru 2016), and intelligence analysis (Toniolo et al. 2015). As in ABA, strict rules capture deductively valid inferences. Extending on ABA, defeasible rules capture presumptive inference, where premises create a presumption supporting the conclusion that can be refuted by contrary evidence. In its general form, ASPIC⁺ allows for arguments that combine both strict and defeasible inference rules as well as preferential information. This generality raises significant challenges for developing effective systems for reasoning in ASPIC⁺. Indeed, practical computational approaches (Snaith and Reed 2012; Visser 2008) to instantiations of ASPIC⁺ remain to-date somewhat underdeveloped (Modgil and Prakken 2018).

In this work, we develop a first ASP-based approach to

reasoning about acceptance in ASPIC⁺. We consider an instantiation of ASPIC⁺ with both strict and defeasible rules composed of atomic sentences, including axioms and ordinary premises, and allowing asymmetric negation. Integral towards a fully declarative approach, we establish foundational underpinnings of phrasing several argumentation semantics in terms of subsets of defeasible elements of this instantiation of ASPIC⁺. Putting our theoretical observations into practice, we detail ASP encodings that allow for reasoning about acceptance in ASPIC⁺ under various central argumentation semantics. We show through an empirical feasibility evaluation that the ASP approach scales up to instances with thousands of sentences. While we do not accommodate for preferences as of yet, our results pave the way for further progress in efficient practical reasoning in ASPIC⁺ more generally. Drawing on known complexity results in the context of ABA and AFs (Dimopoulos, Nebel, and Toni 2002; Dvořák and Dunne 2018), we also explicate complexity results for credulous and skeptical acceptance in the considered instantiation of ASPIC⁺, with completeness for the complexity classes NP, coNP, and Π_2^P depending on the argumentation semantics at hand. Our ASP approach adheres to these complexity requirements, instead of, e.g., first generating a potentially exponential number of arguments and attacks between them before invoking AF reasoning tools.

The rest of this paper is organized as follows. After background on the instantiation of ASPIC⁺ we focus on (Section 2), we rephrase argumentation semantics (in Section 3) towards our answer set programming encodings to deciding acceptance (Section 5) and related complexity results (Section 4). Before conclusions, we provide empirical results on the scalability of the ASP approach (Section 6).

2 The ASPIC⁺ Framework

ASPIC⁺ (Modgil and Prakken 2018; Modgil and Prakken 2013; Prakken 2010) is a general formal framework for argumentation with several ingredients. We assume a language \mathcal{L} composed of atoms x , a contrary function $\bar{\cdot} : \mathcal{L} \rightarrow 2^{\mathcal{L}}$, denoting contraries of $x \in \mathcal{L}$ by $\bar{x} \subseteq \mathcal{L}$. One part of an ASPIC⁺ framework is a knowledge base $\mathcal{K} \subseteq \mathcal{L}$ consisting of a defeasible part (called ordinary premises \mathcal{K}_p) and a non-defeasible part (axioms \mathcal{K}_n).

Definition 1. A knowledge base is a set $\mathcal{K}_n \cup \mathcal{K}_p = \mathcal{K} \subseteq \mathcal{L}$, with disjoint sets \mathcal{K}_n (axioms) and \mathcal{K}_p (ordinary premises).

Another part of ASPIC⁺ is a set of rules over \mathcal{L} , denoted by \mathcal{R} . This set is composed of defeasible rules $a_1, \dots, a_n \Rightarrow b$ and strict rules $a_1, \dots, a_n \rightarrow b$. We denote the set of defeasible rules by \mathcal{R}_d and the set of strict rules by \mathcal{R}_s . When we do not distinguish between strict or defeasible rules, we write $a_1, \dots, a_n \rightsquigarrow b$. A partial function $n : \mathcal{R}_d \rightarrow \mathcal{L}$ names defeasible rules. For a rule $r = a_1, \dots, a_n \rightsquigarrow b$, we denote its head by $head(r) = b$ and its body by $body(r) = \{a_1, \dots, a_n\}$.

The central definition of argumentation theories is based on knowledge bases and rules.

Definition 2. An argumentation theory (AT) is a tuple $(\mathcal{L}, \mathcal{R}, n, \bar{\cdot}, \mathcal{K})$, with a knowledge base $\mathcal{K} \subseteq \mathcal{L}$, rules $\mathcal{R} =$

$\mathcal{R}_d \cup \mathcal{R}_s$ over \mathcal{L} , a contrary function $\bar{\cdot} : \mathcal{L} \rightarrow 2^{\mathcal{L}}$, and a partial function $n : \mathcal{R}_d \rightarrow \mathcal{L}$.

We restrict each part of an AT to be finite. Arguments are constructed from parts of an AT. Intuitively, an argument represents a “derivation tree” starting from elements in the knowledge base and uses rules to derive a conclusion.

Definition 3. Given an AT $T = (\mathcal{L}, \mathcal{R}, n, \bar{\cdot}, \mathcal{K})$, the set of arguments in T is inductively defined as follows.

- If $x \in \mathcal{K}$, then $A = x$ is an argument with $\text{Conc}(A) = x$.
- If A_1, \dots, A_n are arguments, $x_i = \text{Conc}(A_i)$ for $1 \leq i \leq n$, and $(x_1, \dots, x_n \rightsquigarrow x) \in \mathcal{R}$, then $A = A_1, \dots, A_n \rightsquigarrow x$ is an argument with $\text{Conc}(A) = x$.

There are no other arguments.

We use shorthands for the components of arguments.

Definition 4. Let $T = (\mathcal{L}, \mathcal{R}, n, \bar{\cdot}, \mathcal{K})$ be an AT, and A an argument in T .

- If $A = x \in \mathcal{K}$ then $\text{Sub}(A) = \{A\}$ and $\text{Rules}(A) = \emptyset$.
- If $A = A_1, \dots, A_n \rightsquigarrow x$, then $\text{Sub}(A) = \{A\} \cup \bigcup_{i=1}^n \text{Sub}(A_i)$, $\text{TopRule}(A) = (\text{Conc}(A_1), \dots, \text{Conc}(A_n) \rightsquigarrow x)$, and $\text{Rules}(A) = \{\text{TopRule}(A)\} \cup \bigcup_{i=1}^n \text{Rules}(A_i)$.

Further, $\text{Prem}(A) = \text{Sub}(A) \cap \mathcal{K}$, $\text{Prem}_d(A) = \text{Prem}(A) \cap \mathcal{K}_p$, and $\text{DefRules}(A) = \text{Rules}(A) \cap \mathcal{R}_d$.

In words, we use shorthands for the subarguments (Sub) of an argument, the rules and defeasible rules in the argument (Rules and DefRules), the topmost rule (TopRule), the premises of the argument within the knowledge base (Prem), and the ordinary premises (Prem_d). If $A \in \mathcal{K}$, then $\text{TopRule}(A)$ is undefined. We extend the shorthands for a set of arguments \mathcal{A} as $\text{Conc}(\mathcal{A}) = \{\text{Conc}(A) \mid A \in \mathcal{A}\}$ and $\text{TopRule}(\mathcal{A}) = \{\text{TopRule}(A) \mid A \in \mathcal{A}\}$. For each shorthand $f \in \{\text{Sub}, \text{Rules}, \text{DefRules}, \text{Prem}, \text{Prem}_d\}$ returning a set, we define $f(\mathcal{A}) = \bigcup_{A \in \mathcal{A}} f(A)$. An argument A is an immediate subargument of $B = A_1, \dots, A_n \rightsquigarrow x$ if $A \in \{A_1, \dots, A_n\}$. As in (Modgil and Prakken 2018), we focus on finite structures as arguments (i.e., on arguments which are “trees” of finite size), and consider as arguments those arguments A for which $\text{Sub}(A)$ is finite (disallowing infinite chaining of rules, e.g., via $x \rightsquigarrow x$).

Example 1. Consider the example AT $T = (\mathcal{L}, \mathcal{R}, n, \bar{\cdot}, \mathcal{K})$ with $\mathcal{L} = \{a, b, c, w, x, y, z, r_1, r_2, r_3\}$, $\mathcal{K}_p = \{b\}$, $\mathcal{K}_n = \{a, c\}$, and contraries $\bar{r}_2 = \{z\}$, $\bar{b} = \{w\}$, and $\bar{w} = \{x\}$, with $\mathcal{R} = \mathcal{R}_d \cup \mathcal{R}_s$ given by the defeasible rules (names in brackets) $a \Rightarrow w$ (r_1), $c \Rightarrow y$ (r_2), and $x \Rightarrow z$ (r_3), and the only strict rule $b \rightarrow x$. The AT is shown in Figure 1(left) with the arguments it gives rise to. Square brackets denote what is in an argument; solid lines non-defeasible elements (axioms and strict rules); and dashed lines defeasible elements (ordinary premises and defeasible rules).

Conflicts among arguments are represented via attacks between arguments.

Definition 5. Given an AT $T = (\mathcal{L}, \mathcal{R}, n, \bar{\cdot}, \mathcal{K})$ and two arguments A and B in T , argument A attacks argument B iff A undercuts, rebuts, or undermines B , where

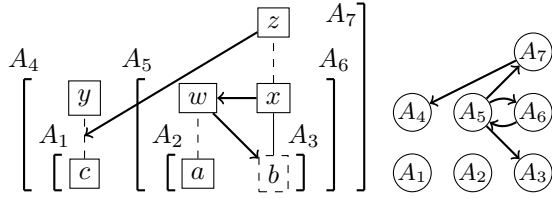


Figure 1: Example AT and corresponding AF

- A undercuts B (on B') iff $\text{Conc}(A) \in \overline{n(r)}$ for some $B' \in \text{Sub}(B)$ such that $\text{TopRule}(B') = r$ is defeasible;
- A rebuts B (on B') iff $\text{Conc}(A) \in \overline{x}$ for some $B' = B_1, \dots, B_n \Rightarrow x \in \text{Sub}(B)$; and
- A undermines B (on x) iff $\text{Conc}(A) \in \overline{x}$ and $x \in \text{Prem}_d(B)$.

In brief, an argument can attack another argument on the defeasible parts of the latter. Ordinary premises can be attacked by undermining, and defeasible rules can be attacked by rebutting the conclusion or undercutting the rule itself. In Figure 1 the attacks among the arguments from Example 1 are depicted as solid arrows. Note that if a subargument is attacked, the superargument is also attacked; these are omitted from Figure 1 (left).

Semantics of ATs are defined via a translation to (abstract) argumentation frameworks (AFs) (Dung 1995). An AF is a pair $F = (\mathcal{A}, \mathcal{D})$ of a set of (abstract) arguments \mathcal{A} and attacks $\mathcal{D} \subseteq \mathcal{A} \times \mathcal{A}$ between arguments. If $(A, B) \in \mathcal{D}$ we say that A attacks B . Similarly, $\mathcal{S} \subseteq \mathcal{A}$ attacks $B \in \mathcal{A}$ if there is an $A \in \mathcal{S}$ with $(A, B) \in \mathcal{D}$. We say that \mathcal{S} defends an argument A if for each $B \in \mathcal{A}$ such that $(B, A) \in \mathcal{D}$, there is a $C \in \mathcal{S}$ such that $(C, B) \in \mathcal{D}$. We consider the AF semantics of conflict-free and admissible sets, and complete, preferred and stable extensions, with the corresponding functions $\sigma \in \{cf, adm, com, prf, stb\}$. A semantics $\sigma(F) \subseteq 2^{\mathcal{A}}$ returns a set of extensions. An extension under a semantics σ is a σ -extension for short.

Definition 6. Given an AF $F = (\mathcal{A}, \mathcal{D})$, a set $\mathcal{E} \subseteq \mathcal{A}$ is conflict-free (in F) if there are no A, B in \mathcal{E} such that $(A, B) \in \mathcal{D}$. The set of all conflict-free sets of F is denoted by $cf(F)$. For an $\mathcal{E} \in cf(F)$, we have that

- $\mathcal{E} \in adm(F)$ iff each $A \in \mathcal{E}$ is defended by \mathcal{E} ;
- $\mathcal{E} \in com(F)$ iff $\mathcal{E} \in adm(F)$ and each A defended by \mathcal{E} is in \mathcal{E} ;
- $\mathcal{E} \in prf(F)$ iff $\mathcal{E} \in adm(F)$ and there is no $\mathcal{T} \in adm(F)$ with $\mathcal{E} \subset \mathcal{T}$; and
- $\mathcal{E} \in stb(F)$ iff \mathcal{E} attacks each argument in $\mathcal{A} \setminus \mathcal{E}$.

ATs can be translated to AFs as follows.

Definition 7. Let $T = (\mathcal{L}, \mathcal{R}, n, \overline{}, \mathcal{K})$ be an AT. An AF $F = (\mathcal{A}, \mathcal{D})$ corresponds to T if \mathcal{A} is the set of all arguments from T and \mathcal{D} the attack relation based on T .

Reasoning on ATs consists of checking whether a queried atom is warranted, by asking (credulously) whether there is a σ -extension having an argument concluding the atom, or (skeptically) whether all σ -extensions have such an argument.

Definition 8. Given an AT $T = (\mathcal{L}, \mathcal{R}, n, \overline{}, \mathcal{K})$ and an AF F corresponding to T , we say that $x \in \mathcal{L}$ is

- skeptically justified in T under semantics σ if in each $\mathcal{E} \in \sigma(F)$ there is an $A \in \mathcal{E}$ with $\text{Conc}(A) = x$;
- credulously justified in T under semantics σ if there is an $\mathcal{E} \in \sigma(F)$ with an $A \in \mathcal{E}$ s.t. $\text{Conc}(A) = x$.

Example 2. Continuing Example 1, an AF corresponding to the AT is shown in Figure 1 (right). The complete extensions of the AF are $\mathcal{E}_1 = \{A_1, A_2\}$, $\mathcal{E}_2 = \{A_1, A_2, A_3, A_6, A_7\}$, and $\mathcal{E}_3 = \{A_1, A_2, A_4, A_5\}$. For instance, $\text{Conc}(\mathcal{E}_1) = \{a, c\}$ (the axioms of the AT) and $\text{Conc}(\mathcal{E}_2) = \{a, b, c, x, z\}$. Since $\{a, c\} \subseteq \text{Conc}(A_1 \cup A_2)$, both a and c are skeptically justified under complete semantics. Since $x \in \text{Conc}(\mathcal{E}_2)$ and $x \notin \text{Conc}(\mathcal{E}_1)$, x is credulously but not skeptically justified under complete semantics. \mathcal{E}_2 and \mathcal{E}_3 are the only preferred (stable) extensions of the AF.

3 Rephrasing Semantics

In general the corresponding AF for an AT might be large in size, which can be a barrier for computation. We approach deciding whether an atom is justified from a different angle, inspired by a similar approach in ABA (Cyras et al. 2018) and studies of general rule-based systems (Amgoud and Besnard 2019): we look at parts of the defeasible elements of an AT, dubbed assumptions, and show that one can restate semantics of ATs in a sufficient manner by imposing conditions on assumptions.

Definition 9. For an AT $T = (\mathcal{L}, \mathcal{R}, n, \overline{}, \mathcal{K})$ we call a pair (P, D) with $P \subseteq \mathcal{K}_p$ and $D \subseteq \mathcal{R}_d$ an assumption in T .

We compare two assumptions (P, D) and (P', D') for an AT T via \sqsubseteq defined by $(P, D) \sqsubseteq (P', D')$ iff $P \subseteq P'$ and $D \subseteq D'$. The strict variant is defined as $(P, D) \sqsubset (P', D')$ iff $(P, D) \sqsubseteq (P', D')$ and $\exists x \in P' \cup D'$ with $x \notin P \cup D$.

Given a set of rules \mathcal{R} and a set of atoms $L \subseteq \mathcal{L}$, we write that $x \in \mathcal{L}$ is derivable from L via \mathcal{R} , denoted by $L \vdash_{\mathcal{R}} x$, if (i) $x \in L$ or (ii) there is a sequence of rules (r_1, \dots, r_n) from \mathcal{R} s.t. $\text{head}(r_n) = x$ and for each rule r_i it holds that each atom in the body of r_i is derived from rules earlier in the sequence or is in L , i.e., $\text{body}(r_i) \subseteq L \cup \bigcup_{j < i} \text{head}(r_j)$.

We extend the derivability notion to assumptions in a straightforward way: given an AT $T = (\mathcal{L}, \mathcal{R}, n, \overline{}, \mathcal{K})$ and an assumption (P, D) in T , we say that from (P, D) one can derive (within T) an atom $x \in \mathcal{L}$, denoted by $(P, D) \vdash_T x$, if $P \cup \mathcal{K}_n \vdash_{D \cup \mathcal{R}_s} x$, i.e., x is derivable from the defeasible elements in the assumption and all non-defeasible parts in the AT. The deductive closure of an assumption in T is then defined as $Th_T(P, D) = \{x \in \mathcal{L} \mid (P, D) \vdash_T x\}$. We say that a rule r is applicable by an assumption (P, D) if $\text{body}(r) \subseteq Th_T(P, D)$, i.e., all elements of the body of r can be derived using the assumption.

There is a natural connection of derivability via assumptions (P, D) and conclusions of arguments composed of elements in the assumption along with strict rules and axioms. For an assumption (P, D) in T and an argument A in T we say that A is based on (P, D) if A uses only defeasible elements from this assumption, i.e., A is based on (P, D) in T if $\text{DefRules}(A) \subseteq D$ and $\text{Prem}_d(A) \subseteq P$.

Proposition 1. Let $T = (\mathcal{L}, \mathcal{R}, n, \neg, \mathcal{K})$ be an AT and (P, D) an assumption in T . There is an argument A based on (P, D) in T with $\text{Conc}(A) = x$ iff $(P, D) \vdash_T x$.

Proof. Define the height of an argument A inductively as $h(A) = 0$ if $A \in \mathcal{K}$ and $h(A) = n + 1$ if $A \notin \mathcal{K}$ and $n = \max\{h(A') \mid A' \text{ immediate subargument of } A\}$. Assume that A is based on (P, D) in T with $\text{Conc}(A) = x$. The claim is proven by induction on the height. For $h(A) = 0$ the claim follows directly ($A \in \mathcal{K}_n \cup P$). Now assume the claim holds for all arguments A with $n = h(A)$. Thus, for any argument A' with $n + 1 = h(A')$, the claim holds for its immediate subarguments, implying the claim for A' (one can derive $\text{Conc}(A')$ from (P, D) via the top rule of A').

For the other direction assume that $(P, D) \vdash_T x$. The case with $x \in \mathcal{K}$ is straightforward. Assume $x \notin \mathcal{K}$. Then there is a sequence of rules (r_1, \dots, r_n) with each rule applicable by earlier rules in the sequence and by P , and $\text{Conc}(r_n) = x$. We show that for each i , $1 \leq i \leq n$ we have an argument A based on (P, D) in T with $\text{Conc}(A) = \text{head}(r_i)$. By an inductive argument, for $i = 1$ it holds that $\text{body}(r_1) \subseteq \mathcal{K}_n \cup P$, implying existence of an A with $\text{Prem}(A) \subseteq \mathcal{K}_n \cup P$, $r_1 \in \mathcal{R}_s \cup D$, and $\text{Conc}(A) = \text{head}(r_1)$. Assume that the claim holds for $i = m$. We show that it holds also for $m + 1$. By definition and assumption, it holds that $\text{body}(r_{m+1}) \subseteq \{\text{head}(r_j) \mid j < m + 1\} \cup P \cup \mathcal{K}_n$, implying that there is a set of arguments \mathcal{S} based on (P, D) in T such that $\text{body}(r_{m+1}) \subseteq \{\text{Conc}(A') \mid A' \in \mathcal{S}\} \cup P \cup \mathcal{K}_n$. \square

The central argumentative concept of attacks can also be defined on assumptions.

Definition 10. Let $T = (\mathcal{L}, \mathcal{R}, n, \neg, \mathcal{K})$ be an AT and (P, D) an assumption in T . We say that (P, D) attacks

- $a \in \mathcal{K}_p$ if $\overline{a} \cap \text{Th}_T(P, D) \neq \emptyset$, and
- an $r \in \mathcal{R}_d$ if
 - $\overline{n(r)} \cap \text{Th}_T(P, D) \neq \emptyset$ or
 - $\overline{\text{head}(r)} \cap \text{Th}_T(P, D) \neq \emptyset$.

We define $\text{att}(P, D) = \{x \in \mathcal{K}_p \cup \mathcal{R}_d \mid (P, D) \text{ attacks } x\}$.

In words, an assumption attacks an ordinary premise if one can derive a contrary of the ordinary premise. A defeasible rule is attacked if either a contrary of the rule (name) can be derived or a contrary of the head can be derived. Attacks via assumptions are then connected to attacks on arguments by the following corollary.

Corollary 2. Let $T = (\mathcal{L}, \mathcal{R}, n, \neg, \mathcal{K})$ be an AT, (P, D) an assumption in T , and $p \in \mathcal{K}_p$ and $r \in \mathcal{R}_d$. Then

- (P, D) attacks p iff there is an argument A based on (P, D) in T with $\text{Conc}(A) \in \overline{p}$,
- (P, D) attacks r iff there is an argument A based on (P, D) in T with $\text{Conc}(A) \in \overline{n(r)} \cup \overline{\text{head}(r)}$, and
- for an argument B in T , (P, D) attacks an $x \in \text{Prem}_d(B) \cup \text{DefRules}(B)$ iff there is an argument A based on (P, D) in T that attacks B .

Proof. This follows from Proposition 1. The first two items follow straightforwardly. Assume that (P, D) attacks an

$x \in \text{Prem}_d(B) \cup \text{DefRules}(B)$ for some argument B in T . If $x \in \mathcal{K}_p$ then $\overline{x} \cap \text{Th}_T(P, D) \neq \emptyset$. If $x \in \mathcal{R}_d$, then either $\text{Th}_T(P, D) \cap \overline{n(x)} \neq \emptyset$ or $\text{Th}_T(P, D) \cap \overline{\text{head}(x)} \neq \emptyset$. Consider then the following three associated cases. If (i) $x \in \mathcal{K}_p$, by Proposition 1 there is an argument A based on (P, D) with $\text{Conc}(A) \in \overline{x}$ that undermines B on x . If (ii) $x \in \mathcal{R}_d$ and $\text{Th}_T(P, D) \cap \overline{n(x)} \neq \emptyset$, there is an argument A based on (P, D) in T with $\text{Conc}(A) \in \overline{n(x)}$ that undercuts B on a $B' \in \text{Sub}(B)$ (since $x \in \text{DefRules}(B)$, there must be a subargument B' with $\text{TopRule}(B') = x$). If (iii) $x \in \mathcal{R}_d$ and $\overline{\text{head}(x)} \cap \text{Th}_T(P, D) \neq \emptyset$, there is an argument A based on (P, D) with $\text{Conc}(A) \in \overline{\text{head}(x)}$ that rebuts B on some $B' \in \text{Sub}(B)$ (since A concludes a contrary of $\text{Conc}(B')$ and $\text{TopRule}(B')$ is defeasible). Analogously for the other direction: A attacks B if either A undermines, undercuts, or rebuts B (potentially on some $B' \in \text{Sub}(B)$), implying in all cases that $\text{Conc}(A)$ is a corresponding contrary and that from (P, D) one can derive this contrary and that (P, D) attacks an $x \in \text{Prem}(B) \cup \text{DefRules}(B)$ as defined in Definition 10. \square

For intuition on defense on assumptions, an assumption (P, D) defends a defeasible element x if from the AT excluding all defeasible elements attacked by (P, D) no attacks target x (i.e., all attacks on x are attacked by (P, D)).

Definition 11. Let $T = (\mathcal{L}, \mathcal{R}, n, \neg, \mathcal{K})$ be an AT and (P, D) be an assumption in T . Assumption (P, D) defends an $x \in \mathcal{K}_p \cup \mathcal{R}_d$ if assumption (P', D') does not attack x with

- $P' = \mathcal{K}_p \setminus \text{att}(P, D)$ and
- $D' = \mathcal{R}_d \setminus \text{att}(P, D)$.

Let $\text{def}(P, D) = (\{p \in \mathcal{K}_p \mid (P, D) \text{ defends } p\}, \{r \in \mathcal{R}_d \mid (P, D) \text{ defends } r\})$.

Example 3. Consider the assumptions $M_1 = (\emptyset, \emptyset)$, $M_2 = (\{b\}, \{r_3\})$, and $M_3 = (\{b\}, \{r_1\})$ of the AT from Example 1. Now $\text{att}(M_1) = \emptyset$ (no contrary derivable from axioms and strict rules), $\text{att}(M_2) = \{r_1, r_2\}$ (since $\overline{w}, \overline{r_2} \in \text{Th}_T(M_2)$ and r_1 concludes w), and $\text{att}(M_3) = \{b, r_1\}$. Regarding defense, $\text{def}(M_1) = (\emptyset, \{r_3\})$ since r_3 is never attacked by an assumption (no argument undercuts or rebuts arguments based on r_3). Further, $\text{def}(M_2) = (\{b\}, \{r_3\})$ (it holds that $(\emptyset, \{r_1\})$ attacks b , but r_1 is attacked by M_2) and $\text{def}(M_3) = (\{b\}, \{r_1, r_2, r_3\})$.

Based on attacks and defense on assumptions, we next state conditions on assumptions reflecting the conditions of the corresponding semantics of ATs.

Definition 12. Let $T = (\mathcal{L}, \mathcal{R}, n, \neg, \mathcal{K})$ be an AT. Assumption (P, D) in T is conflict-free (in T) if (P, D) does not attack an $x \in P \cup D$. Further, a conflict-free assumption (P, D) in T with all rules $r \in D$ applicable by (P, D) is

- admissible (in T) iff $(P, D) \sqsubseteq \text{def}(P, D)$;
- complete (in T) iff (P, D) is admissible in T and $\text{def}(P, D) = (P, D')$ with each $r \in D' \setminus D$ not applicable by (P, D') ;
- stable (in T) iff $\mathcal{K}_p = (\text{att}(P, D) \cup P)$ and no rule $r \in \mathcal{R}_d \setminus (\text{att}(P, D) \cup D)$ is applicable by $(\mathcal{K}_p \setminus \text{att}(P, D), \mathcal{R}_d \setminus \text{att}(P, D))$; and

- *preferred (in T) iff (P, D) is \sqsubseteq -maximal admissible in T .*

In words, an assumption is conflict-free if the assumption does not attack itself (on an ordinary premise or a defeasible rule). We require for a σ -assumption that each rule is applicable (the body of each rule is derivable). An assumption is admissible if it defends each of its defeasible elements, and complete if additionally everything defended by the assumption is included except non-applicable rules (in $D' \setminus D$). Stability requires that each ordinary premise is attacked or in the assumption, while a defeasible rule is either attacked, in the assumption, or non-applicable by unattacked defeasible elements. Finally, preferred assumptions are \sqsubseteq -maximal admissible assumptions. All these definitions reflect criteria of argumentation semantics (Definition 6). However, special care is required for the components of an AT (e.g., non-applicability of rules).

Example 4. *The complete assumptions of the AT of Example 1 are (\emptyset, \emptyset) , $(\{b\}, \{r_3\})$, and $(\emptyset, \{r_1, r_2\})$. To see that the first two are complete, consider the corresponding attack and defense in Example 3. Note that each assumption defends r_3 . However, (\emptyset, \emptyset) , for instance, does not include r_3 ; a complete (admissible) assumption set may not contain a non-applicable defeasible rule, and completeness does not require to include non-applicable defended rules.*

Before establishing a connection of σ -assumptions and σ -extensions of an AF corresponding to a particular AT, we show basic properties of assumptions. Similarly as in AFs, each stable assumption is admissible.

Proposition 3. *Let $T = (\mathcal{L}, \mathcal{R}, n, \neg, \mathcal{K})$ be an AT. Each stable assumption in T is admissible in T .*

Proof. Let (P, D) be stable in T . Suppose (P, D) is not admissible in T . Then there is an $x \in P \cup D$ not defended by (P, D) , and $(P', D') = (\mathcal{K}_p \setminus \text{att}(P, D), \mathcal{R}_d \setminus \text{att}(P, D))$ attacks x . By definition, $P' \subseteq P$ and each $r \in \mathcal{R}_d \setminus \text{att}(P, D)$ is either in D or not applicable by (P', D') . In the latter case r does not occur in any derivation from (P', D') , thus, (P', D') attacks x iff $(P', D' \setminus \{r\})$ attacks x . Let D'' be D' without rules not applicable by (P', D') . We have (P', D'') attacks x and $D'' \subseteq D$, implying that (P, D) attacks x , a contradiction. \square

The next lemma states that if one collects all ordinary premises and defeasible rules of a complete extension of an AF corresponding to a particular AT, then this complete extension contains all arguments based on these defeasible elements. Intuitively, complete extensions include all arguments based on the same defeasible parts, and, in turn, contain each “derivation” (via arguments) based on these defeasible parts.

Lemma 4. *Let $T = (\mathcal{L}, \mathcal{R}, n, \neg, \mathcal{K})$ be an AT and F an AF corresponding to T . If \mathcal{E} is complete in F then each argument based on $(\text{Prem}_d(\mathcal{E}), \text{DefRules}(\mathcal{E}))$ is part of \mathcal{E} .*

Proof. Suppose that \mathcal{E} is complete in F and that there is an argument $A \notin \mathcal{E}$ based on $(P, D) = (\text{Prem}_d(\mathcal{E}), \text{DefRules}(\mathcal{E}))$ in T . Hence there is an argument B in T s.t. B attacks A and \mathcal{E} does not attack B (otherwise \mathcal{E} would not be complete). It holds that B undermines,

undercuts, or rebuts A . Thus, either B undermines A on some $p \in \text{Prem}_d(A)$, or B undercuts or rebuts A on some $A' \in \text{Sub}(A)$ with the top rule of A' being defeasible. By construction, for each $x \in P \cup D$, there is an argument $C \in \mathcal{E}$ with $x \in \text{Prem}_d(C) \cup \text{DefRules}(C)$. This implies that B attacks a $C \in \mathcal{E}$ (since A and C are based on (P, D) and share the same “opening” for an attack from B). By admissibility of \mathcal{E} we have \mathcal{E} attacks B , a contradiction. \square

We next state the main formal results connecting reasoning via assumptions and extensions. Intuitively, for each σ -extension of an AF corresponding to a AT there is a corresponding σ -assumption based on the same defeasible elements, and vice versa. Proofs for other semantics apart from admissibility are in Appendix A.

Theorem 5. *Let $T = (\mathcal{L}, \mathcal{R}, n, \neg, \mathcal{K})$ be an AT, $\sigma \in \{\text{adm}, \text{com}, \text{prf}, \text{stb}\}$, and F an AF corresponding to T .*

- *If (P, D) is a σ -assumption in T , then $\mathcal{E} = \{A \mid A \text{ based on } (P, D) \text{ in } T\}$ is a σ -extension of F .*
- *If \mathcal{E} is a σ -extension of F , then (P, D) is a σ -assumption of T with $P = \text{Prem}_d(\mathcal{E})$ and $D = \text{DefRules}(\mathcal{E})$.*

Proof for admissibility. Let $\sigma = \text{adm}$. Let (P, D) be admissible in T and $\mathcal{E} = \{A \mid A \text{ based on } (P, D) \text{ in } T\}$. Suppose \mathcal{E} is not admissible in F . If \mathcal{E} is not conflict-free, then there are arguments $A, B \in \mathcal{E}$ s.t. A attacks B in F . Due to Corollary 2, then (P, D) attacks some $x \in \text{Prem}_d(B) \cup \text{DefRules}(B) \subseteq P \cup D$, implying that (P, D) is not conflict-free in T . Thus \mathcal{E} is conflict-free in F . In order for \mathcal{E} to violate admissibility, it must hold that there is an argument B in T that attacks an argument $A \in \mathcal{E}$ and \mathcal{E} does not attack B . For any $C \in \mathcal{E}$ it holds that C is based on (P, D) in T . Thus, \mathcal{E} does not attack B iff (P, D) does not attack any $x \in \text{Prem}_d(B) \cup \text{DefRules}(B)$. This implies that $(\mathcal{K}_p \setminus \text{att}(P, D), \mathcal{R}_d \setminus \text{att}(P, D))$ attacks a $y \in P \cup D$, violating admissibility of (P, D) .

For the other direction, assume that \mathcal{E} is admissible in F , and let (P, D) be as above. For each $x \in P \cup D$ there is an argument $B \in \mathcal{E}$ with $x \in \text{Prem}_d(B) \cup \text{DefRules}(B)$. Since each $r \in D$ is part of at least one argument, r is applicable by (P, D) . Suppose that (P, D) is not conflict-free, implying (P, D) attacks some $x \in P \cup D$. Thus there is an argument $B \in \mathcal{E}$ based on (P, D) with $x \in \text{Prem}_d(B) \cup \text{DefRules}(B)$. By Corollary 2, there is an argument A based on (P, D) with A attacking B . Due to \mathcal{E} being admissible, it holds that there is a $C \in \mathcal{E}$ that attacks A (defending B). Thus, $\text{Conc}(C)$ (i) is in \overline{p} for some $p \in \text{Prem}_d(A)$ (if C undermines A), (ii) is in $\overline{n(r)}$ for some $r \in \text{DefRules}(A)$ (if C undercuts A), or (iii) is in $\overline{\text{head}(r)}$ for some $r \in \text{DefRules}(A)$ (if C rebuts A). Let y be p in case C undermines A on p , or, otherwise, let y be the defeasible rule r in case C undercuts or rebuts A on $A' \in \text{Sub}(A)$ with r being the top rule of A' . Since A is based on (P, D) , there is a $E \in \mathcal{E}$ with $y \in \text{Prem}_d(E) \cup \text{DefRules}(E)$. Hence C also attacks argument E in T . Thus both C and E are in \mathcal{E} , and hence \mathcal{E} is not conflict-free. Suppose that (P, D) is conflict-free but not admissible. Then (P, D) is not admissible iff $(P, D) \not\sqsubseteq \text{def}(P, D)$ iff there is an

$x \in P \cup D$ that (P, D) does not defend iff $(P', D') = (\mathcal{K}_p \setminus \text{att}(P, D), \mathcal{R}_d \setminus \text{att}(P, D))$ attacks an $x \in P \cup D$. By Corollary 2 and the above, there is an argument B based on (P', D') that attacks some argument $A \in \mathcal{E}$. Due to \mathcal{E} being admissible there is a $C \in \mathcal{E}$ that attacks B . However, this means that C is based on (P, D) and that C attacks B with $\text{Prem}_d(B) \cup \text{DefRules}(B) \subseteq P' \cup D'$, implying that (P, D) attacks some $y \in P' \cup D'$, a contradiction. \square

Finally, we tie credulous and skeptical reasoning on ATs to assumptions, enabling deciding these reasoning tasks solely based on assumptions. First, we note that if an atom $x \in \mathcal{L}$ is credulously justified under admissibility in an AT T , then x is also credulously justified under complete and preferred semantics. To see this, the former implies existence of an admissible set \mathcal{E} in the AF F corresponding to T with $x \in \text{Conc}(\mathcal{E})$. Thus there is a complete and preferred extension \mathcal{E}' with $\mathcal{E} \subseteq \mathcal{E}'$. Similarly, if x is credulously justified under preferred (complete) semantics, a witnessing preferred (complete) extension \mathcal{E} contains an argument concluding x , and \mathcal{E} is admissible (complete).

Proposition 6. *Let $T = (\mathcal{L}, \mathcal{R}, n, \neg, \mathcal{K})$ be an AT, $\sigma \in \{\text{adm}, \text{com}, \text{prf}, \text{stb}\}$, $\sigma' \in \{\text{stb}, \text{prf}\}$, and $x \in \mathcal{L}$. Then*

- *x is credulously justified in T under σ iff there is a σ -assumption (P, D) in T with $x \in \text{Th}_T(P, D)$, and*
- *x is skeptically justified in T under σ' iff in all σ' -assumptions (P, D) in T we find $x \in \text{Th}_T(P, D)$.*

Proof. Let F be an AF corresponding to T . Let x be credulously justified in T under σ . By definition, there is a σ -extension $\mathcal{E} \in \sigma(F)$ with some $A \in \mathcal{E}$ and $\text{Conc}(A) = x$. By Theorem 5, $(P, D) = (\text{Prem}_d(\mathcal{E}), \text{DefRules}(\mathcal{E}))$ is a σ -assumption in T . Since A is based on (P, D) , it holds that $x \in \text{Th}_T(P, D)$. For the other direction, if there is a σ -assumption (P, D) with $x \in \text{Th}_T(P, D)$, then by Theorem 5 there is a σ -extension $\mathcal{E} = \{A \mid A \text{ based on } (P, D)\}$ of F . If $x \in \text{Th}_T(P, D)$ then there is an argument $A \in \mathcal{E}$ with $\text{Conc}(A) = x$ (due to Proposition 1 and \mathcal{E} containing all arguments based on (P, D)).

For skeptical reasoning, let x be skeptically justified under σ' in T . Suppose that there is a σ' -assumption (P, D) with $x \notin \text{Th}_T(P, D)$. Then by Theorem 5 there is a σ' -extension $\mathcal{E} = \{A \mid A \text{ based on } (P, D)\}$ in F . If $x \notin \text{Th}_T(P, D)$, then there is no argument $A \in \mathcal{E}$ with $\text{Conc}(A) = x$, a contradiction. For the other direction, assume that for all σ' -assumptions (P, D) we have $x \in \text{Th}_T(P, D)$. Suppose that x is not skeptically justified under σ' in T . Then there is a σ' -extension \mathcal{E} of F s.t. no argument in \mathcal{E} concludes x . By Theorem 5, it holds that $(P', D') = (\text{Prem}_d(\mathcal{E}), \text{DefRules}(\mathcal{E}))$ is a σ' -assumption in T . By Lemma 4 and \mathcal{E} being complete, \mathcal{E} contains each argument A based on (P', D') . Since (P', D') is a σ' -assumption, then $x \in \text{Th}_T(P', D')$. Thus, there is an argument A based on (P', D') that concludes x , contradicting that no argument in \mathcal{E} concludes x . \square

4 Complexity of Reasoning in ASPIC⁺

Stating semantics of ATs on assumptions allows for making observations on the complexity of credulous and skeptical reasoning in ATs. In particular, assumptions allow for

working on structures that are inherently bound polynomially by the input instance, whereas translation to AFs may not be bound in such a way (ATs yielding exponential-size AFs exist). Our results hence enable both novel algorithms and complexity results (membership) for ATs. For hardness results, translations of corresponding problems exist for assumption-based argumentation (Modgil and Prakken 2018; Heyninck and Straßer 2016) and AFs (Maher 2017). Here we present a direct reduction from reasoning tasks on AFs to establish hardness.

Proposition 7. *For a given AT, deciding credulous justification is NP-complete under admissible, complete, stable, and preferred semantics. Deciding skeptical justification is coNP-complete under stable semantics and Π_2^P -complete under preferred semantics.*

Proof. For membership, first note that checking if a (P, D) pair is a σ -assumption for all semantics, except preferred, is polytime (Definition 12): computing derivations, attacks, and defended elements from assumptions can be done in polynomial time by chaining applicable rules. Thus by non-deterministic construction of a (P, D) -pair (both sets are bounded by the size of an input AT) and subsequent check of the semantics conditions yields membership for credulous reasoning. Recall that credulous reasoning under *adm*, *com*, and *prf* coincides. For the complement of skeptical reasoning, non-deterministically construct a (P, D) -pair and verify that this assumption is stable and does not derive the queried atom. For preferred semantics, in order to verify, first check admissibility, and then ask an NP oracle whether there is a (P', D') assumption that is admissible in T with $(P, D) \sqsubset (P', D')$. If such an assumption does not exist, the first assumption is preferred.

For hardness, reduce from the corresponding problems on AFs. Given an AF $F = (\mathcal{A}, \mathcal{D})$, construct the AT $T = (\mathcal{A}, \emptyset, n, \neg, \mathcal{K})$ with $\mathcal{K}_p = \mathcal{A}$, and $\overline{\mathcal{A}} = \{B \mid (B, A) \in \mathcal{D}\}$ for each $A \in \mathcal{A}$. Now $A = p$ with $p \in \mathcal{K}_p$ are the only arguments in T . There is a bijection between \mathcal{A} and the set of arguments \mathcal{A}' in the AF $F' = (\mathcal{A}', \mathcal{D}')$ that corresponds to T . Further, $(A, B) \in \mathcal{D}$ iff $(A, B) \in \mathcal{D}'$. As A is credulously (skeptically) accepted in F under σ iff A is credulously (skeptically) justified in T (via F') under σ , hardness follows (Dvořák and Dunne 2018). \square

5 ASPIC⁺ Reasoning via ASP

ASP. An ASP π consists of rules r of the form $b_0 \leftarrow b_1, \dots, b_k, \text{not } b_{k+1}, \dots, \text{not } b_m$, where each b_i is an atom. A rule is positive if $k = m$ and a fact if $m = 0$. A literal is an atom b_i or *not* b_i . A rule without head b_0 is a constraint and a shorthand for $a \leftarrow b_1, \dots, b_k, \text{not } b_{k+1}, \dots, \text{not } b_m, \text{not } a$ for a fresh a . An atom b_i is $p(t_1, \dots, t_n)$ with each t_j either a constant or a variable. An answer set program is ground if it is free of variables. For a non-ground program, GP is the set of rules obtained by applying all possible substitutions from the variables to the set of constants appearing in the program. An interpretation I , i.e., a subset of all the ground atoms, satisfies a positive rule $r = h \leftarrow b_1, \dots, b_k$ iff all positive body elements b_1, \dots, b_k being in I implies

Listing 1: Module π_{common}

```

1 in(X) ← axiom(X).
2 in(X) ← premise(X), not out(X).
3 out(X) ← premise(X), not in(X).
4 in(X) ← strict_head(X, _).
5 in(X) ← head(X, _), not out(X).
6 out(X) ← head(X, _), not in(X).
7 supported(X) ← axiom(X).
8 supported(X) ← premise(X), in(X).
9 supported(X) ← head(R, X), applicable_by_in(R).
10 supported(X) ← strict_head(R, X), applicable_by_in(R).
11 applicable_by_in(R) ← in(R), head(R, _),
    supported(X) : body(R, X).
12 applicable_by_in(R) ← strict_head(R, _), supported(X) :
    strict_body(R, X).
13 ← in(R), not applicable_by_in(R), head(R, _).
14 defeated(X) ← supported(Y), contrary(X, Y), premise(X).
15 defeated(X) ← supported(Y), contrary(X, Y), head(X, _).
16 defeated(X) ← head(X, S), supported(Y), contrary(S, Y).
17 ← in(X), defeated(X).
18 supported_by_undefeated(X) ← axiom(X).
19 supported_by_undefeated(X) ← premise(X),
    not defeated(X).
20 supported_by_undefeated(X) ← head(R, X),
    applicable_by_undefeated(R).
21 supported_by_undefeated(X) ← strict_head(R, X),
    applicable_by_undefeated(R).
22 applicable_by_undefeated(R) ← not defeated(R), head(R, _),
    supported_by_undefeated(X) : body(R, X).
23 applicable_by_undefeated(R) ← strict_head(R, _),
    supported_by_undefeated(X) : strict_body(R, X).

```

that the head atom is in I . For a program π consisting only of positive rules, let $Cl(\pi)$ be the uniquely determined interpretation I that satisfies all rules in π and no subset of I satisfies all rules in π . Interpretation I is an answer set of a ground program π if $I = Cl(\pi^I)$ where $\pi^I = \{(h \leftarrow b_1, \dots, b_k) \mid (h \leftarrow b_1, \dots, b_k, not\ b_{k+1}, \dots, not\ b_m) \in \pi, \{b_{k+1}, \dots, b_m\} \cap I = \emptyset\}$ is the reduct; and of a non-ground program π if I is an answer set of GP of π .

ASP Encodings. We give each $r \in \mathcal{R}$ in an AT a name (recall that n might be partial, and only for defeasible rules). Let \mathcal{R}'_d be the set of defeasible rules n is defined on. We define an extension n' of n as $n'(r) = n(r)$ if $r \in \mathcal{R}'_d$, and for the remaining rules $r_i \in \mathcal{R} \setminus \mathcal{R}'_d = \{r_1, \dots, r_n\}$ we define $n'(r_i) = i$. We represent an AT $T = (\mathcal{L}, \mathcal{R}, n, \bar{\cdot}, \mathcal{K})$ in ASP as the set of facts $AT(T)$, defined as follows:

$$\begin{aligned}
& \{\text{axiom}(a). \mid a \in \mathcal{K}_n\} \cup \{\text{premise}(a). \mid a \in \mathcal{K}_p\} \cup \\
& \{\text{head}(n'(r), b). \mid r \in \mathcal{R}_d, b = \text{head}(r)\} \cup \\
& \{\text{body}(n'(r), b). \mid r \in \mathcal{R}_d, b \in \text{body}(r)\} \cup \\
& \{\text{strict_head}(n'(r), b). \mid r \in \mathcal{R}_s, b = \text{head}(r)\} \cup \\
& \{\text{strict_body}(n'(r), b). \mid r \in \mathcal{R}_s, b \in \text{body}(r)\} \cup \\
& \{\text{contrary}(a, b). \mid b \in \bar{a}, a \in \mathcal{L}\}.
\end{aligned}$$

We present ASP encodings for reasoning in ASPIC⁺ based on the restatement of semantics in terms of assumptions (i.e., pairs of premises and defeasible rules), recall Definition 12

Listing 2: Module Δ_{stb}

```

1 ← out(X), not defeated(X), premise(X).
2 ← out(R), applicable_by_undefeated(R).

```

and Theorem 5. For our encodings π_σ of semantics $\sigma \in \{\text{adm}, \text{com}, \text{stb}, \text{prf}\}$ it holds that (P, D) is a σ -assumption in T iff there is an answer set M of $AT(T) \cup \pi_\sigma$ with $P \cup D = \{p \in (\mathcal{K}_p \cup \mathcal{R}_d) \mid \text{in}(p) \in M\}$. The encodings enable credulous (skeptical) reasoning by specifying a query atom $a \in \mathcal{L}$ as **query**(a) and adding a constraint enforcing that the atom must (must not) be derivable from (P, D) .

The subprogram π_{common} is common to all the semantics. Lines 1–6 encode a non-deterministic guess of a subset of premises and defeasible rules that are **in** and **out**, with axioms and strict rules always considered **in**. Lines 7–12 encode what is **supported** by (i.e., derivable from) the guessed **in** elements. The base cases are on Lines 7–8: axioms and premises that are **in** are **supported**. Lines 9–10 encode that an atom is **supported** if it is a head of a strict or defeasible rule that is applicable by the **in** elements. A defeasible rule is applicable if the rule is **in** and all its body elements are **supported** (Line 11), with the body encoded here as a list of **supported**(X) for all X in the body using the conditional literal “:”. A strict rule is applicable when all body elements are **supported** (Line 12). Line 13 encodes the restriction that each defeasible rule that is **in** must be applicable by **in**. The three types of attack from **in** are encoded on Lines 14–16, encoding what is **defeated** by **in**, with Line 17 enforcing conflict-freeness of **in**. Lines 18–23 construct the set of atoms supported by premises and rules that are not attacked by **in** in a similar manner as for **in**.

To extend π_{common} to correspond to a particular semantics, we employ Definition 12. The subprogram Δ_{stb} extends to stable semantics: there are no premises that are not **in** and not **defeated** by **in** (Line 1), and there are no rules that are not **in** but are applicable by the elements that are not attacked by **in** (Line 2). The encoding for stable semantics is $\pi_{stb} = \pi_{common} \cup \Delta_{stb}$.

Admissibility is encoded as $\pi_{adm} = \pi_{common} \cup \Delta_{adm}$. In Δ_{adm} Lines 1–3 encode elements attacked by the elements not attacked by **in**. Line 4 states that no such elements can be **in** (since this would indicate an undefended attack to **in**).

Complete semantics is encoded as $\pi_{com} = \pi_{adm} \cup \Delta_{com}$. Lines 1–6 of Δ_{com} compute atoms that are derivable from

Listing 3: Module Δ_{adm}

```

1 defeated_by_undefeated(X) ← supported_by_undefeated(Y),
    contrary(X, Y), premise(X).
2 defeated_by_undefeated(X) ← supported_by_undefeated(Y),
    contrary(X, Y), head(X, _).
3 defeated_by_undefeated(X) ← head(X, S),
    supported_by_undefeated(Y), contrary(S, Y).
4 ← in(X), defeated_by_undefeated(X).

```

Listing 4: Module π_{com}

```

1 supported_by_defended(X)  $\leftarrow$  axiom(X).
2 supported_by_defended(X)  $\leftarrow$  premise(X), not
   defeated_by_undefeated(X).
3 supported_by_defended(X)  $\leftarrow$  head(R,X),
   applicable_by_defended(R).
4 supported_by_defended(X)  $\leftarrow$  strict_head(R,X),
   applicable_by_defended(R).
5 applicable_by_defended(R)  $\leftarrow$ 
   not defeated_by_undefeated(R), head(R,-),
   supported_by_defended(X) : body(R,X).
6 applicable_by_defended(R)  $\leftarrow$  strict_head(R,-),
   supported_by_defended(X) : strict_body(R,X).
7  $\leftarrow$  out(X), not defeated_by_undefeated(X), premise(X).
8  $\leftarrow$  out(R), applicable_by_defended(R).

```

the elements that are not attacked by the set of elements that are attacked by **in**. In other words, the set of atoms defended by **in** is computed. Lines 7–8 enforce that **in** is complete: there can not be premises that are **out** but defended by **in** (Line 7), and no rule outside **in** may be applicable by the elements defended by **in** (Line 8).

Recall that preferred assumptions are the \sqsubseteq -maximal admissible assumptions. Thus preferred assumptions can be computed with the ASP-based system Asprin (Brewka et al. 2015) by adding a constraint that only subset-maximal answers with respect to **in** are returned (as supported by Asprin) on the encoding π_{adm} .

6 Experiments

We study the scalability of our ASP-based approach to deciding acceptance in ASPIC⁺ using the state-of-the-art ASP system Clingo v5.4.0 (Gebser et al. 2018) (with default setting) as the ASP solver and Asprin v3.1.0 (Brewka et al. 2015) for preferred semantics. The experiments were run on Intel Xeon E5-2680 v4 2.4-GHz, 256-GB RAM computers under a per-instance 600-s time and 8-GB memory limit.

We generated benchmark instances for the evaluation as follows. In the following by atoms we mean members of \mathcal{L} excluding the names for defeasible rules. Let N be the number of atoms. We consider ATs with up to $N = 5500$ atoms, with a stepping of 500 starting from 1000 atoms. For each fixed N , we generated 25 frameworks and selected one queried non-premise atom per framework to obtain 25 benchmark instances for each N . In the frameworks, 25% of rules are defeasible and 25% of atoms and defeasible rules have a contrary. For the number of axioms, we considered the values 0.5%, 1%, 5% and 10% of all atoms, and let 20% of atoms be premises. For each non-premise atom, the number of rules deriving the atom was chosen at random from $[1, 15]$ and the number of atoms in the body of each rule body was chosen at random from $[1, 15]$.

Table 1 gives the number of timeouts observed per N with the mean runtimes (timeouts included as 600 s) in parentheses, with 25 instances per N with 1% atoms being axioms for different semantics and reasoning modes. We observe that the approach enables solving a majority of the instances

N	#timeouts (mean runtime over all instances (s))				
	<i>adm cred</i>	<i>com cred</i>	<i>stb cred</i>	<i>stb skept</i>	<i>prf enum</i>
1000	0 (1)	0 (1)	0 (1)	0 (1)	0 (15)
1500	0 (18)	0 (16)	0 (22)	0 (28)	3 (243)
2000	0 (75)	2 (108)	2 (124)	2 (100)	13 (446)
2500	5 (179)	7 (235)	5 (197)	4 (150)	15 (499)
3000	8 (248)	9 (273)	8 (242)	6 (211)	20 (537)
3500	16 (421)	19 (501)	15 (374)	14 (374)	24 (590)
4000	17 (447)	20 (541)	20 (481)	18 (472)	25 (600)
4500	17 (463)	19 (501)	16 (410)	18 (437)	25 (600)
5000	23 (563)	24 (594)	17 (410)	17 (410)	25 (600)
5500	21 (523)	23 (553)	21 (505)	22 (529)	24 (592)

Table 1: Timeouts and mean runtimes with 1% axioms.

up to $N = 3000$ atoms for all considered acceptance problems, with some instances being solved up to $N = 5500$ (and even beyond, with some fluctuations expectedly due to randomness in the instance generation). Figure 2 shows the impact of the relative number of axioms on scalability for *adm*: as the number of axioms is increased (to 10%), the mean runtimes drops, suggesting that small numbers of axioms result in empirically harder instances for the approach.

7 Conclusions

We developed a first ASP-based approach to deciding acceptance in an instantiation of the rule-based structured argumentation framework ASPIC⁺ with both strict and defeasible rules composed of atomic sentences, including axioms and ordinary premises, and allowing asymmetric negation. We rephrased argumentation semantics as subsets of defeasible elements, yielding direct ASP encodings that avoid generating the potentially exponentially-many arguments and attacks between the arguments to decide acceptance. We detailed how credulous and skeptical acceptance for several NP-hard semantics (admissibility, complete, stable, and preferred) are captured via ASP. Empirically, the approach can scale to thousands of atoms. Rephrasing of semantics also yielded complexity results for the considered ASPIC⁺ instantiation. Natural next steps include generalizing the approach to accommodate preferences towards cap-

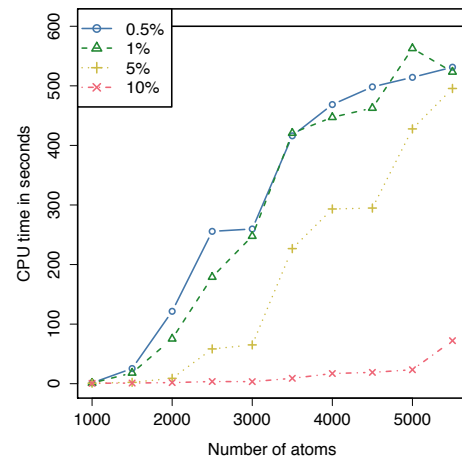


Figure 2: Impact of number of axioms on mean runtimes.

turing ASPIC⁺ more generally and evaluating the approach against other approaches to reasoning in ASPIC⁺.

A Proofs

Proof of Theorem 5. Complete semantics. Let $\sigma = \text{com}$. Let (P, D) be complete in T and \mathcal{E} as defined above. By the proof above, \mathcal{E} is admissible. Suppose \mathcal{E} is not complete in F . Thus, there is an argument A defended by \mathcal{E} and $A \notin \mathcal{E}$, implying that there is an $x \in \text{Prem}_d(A) \cup \text{DefRules}(A)$ with $x \notin P \cup D$. Let $(P', D') = \text{def}(P, D)$. Either (i) $x \notin P \cup D'$ or (ii) $x \in D' \setminus D$. In the latter case, since $x \in \text{DefRules}(A)$, $\text{body}(x) \not\subseteq \text{Th}_T(P, D')$. Thus, for A to be an argument, A cannot be based on (P, D') , implying that $(\text{Prem}_d(A) \cup \text{DefRules}(A)) \not\subseteq (P \cup D')$ (some element outside P and D' is part of A), implying the first case. Thus (P, D) does not defend x , $(P'', D'') = (\mathcal{K}_p \setminus \text{att}(P, D), \mathcal{R}_d \setminus \text{att}(P, D))$ attacks x , and there is an argument B based on (P'', D'') that attacks A (Corollary 2). Since \mathcal{E} defends A , there is an argument $C \in \mathcal{E}$ with C attacking B . However, C is based on (P, D) , implying that (P, D) attacks some $y \in \text{Prem}_d(B) \cup \text{DefRules}(B) \subseteq P'' \cup D''$, a contradiction.

For the other direction, assume that \mathcal{E} is complete in F and (P, D) is not complete in T . By the above, (P, D) is admissible in T . By Lemma 4, \mathcal{E} contains each argument based on (P, D) . For $(P', D') = \text{def}(P, D)$ we have (i) $P \subset P'$ or (ii) $P = P'$ and there is an $r \in D' \setminus D$ that is applicable by (P, D') . Suppose case (i) holds and let $p \in P' \setminus P$. Then (P, D) defends p . By construction, argument $A = p$ is not part of \mathcal{E} . As \mathcal{E} is complete, there is an argument B that attacks A ($\text{Conc}(B) \in \bar{p}$) and \mathcal{E} does not attack B . There is no argument based on (P, D) in T that attacks B iff (P, D) does not attack any $x \in \text{Prem}_d(B) \cup \text{DefRules}(B)$ (Corollary 2). Thus (P, D) does not attack any $x \in \text{Prem}_d(B) \cup \text{DefRules}(B)$. Since additionally $(\text{Prem}_d(B), \text{DefRules}(B))$ does attack p , we have that (P, D) does not defend p : neither $\text{Prem}_d(B)$ nor $\text{DefRules}(B)$ is attacked by (P, D) , however $(\text{Prem}_d(B), \text{DefRules}(B))$ attacks p , implying that (P, D) does not defend p . If case (ii) holds, $P = P'$ and there is an $r \in D' \setminus D$ applicable by (P, D') . Then there is an argument A based on (P, D') with $r \in \text{DefRules}(A)$. As all arguments in \mathcal{E} are based on $(P, D) \sqsubset (P, D')$, we have $A \notin \mathcal{E}$. Thus there is an argument B in T that attacks A and \mathcal{E} does not attack B . As above, then (P, D) does not attack any $x \in \text{Prem}_d(B) \cup \text{DefRules}(B)$. If B undermines A on p , then B attacks \mathcal{E} (since $p \in P$), if B rebuts or undercuts A on A' with $\text{TopRule}(A') = r'$, then B attacks \mathcal{E} if $r' \in D$. Since \mathcal{E} does not attack B , B rebuts or undercuts A on A' with the top rule r' of A' being in $D' \setminus D$. As above, (P, D) does not defend $r' \in D'$: $(\mathcal{K}_p \setminus \text{att}(P, D), \mathcal{R}_d \setminus \text{att}(P, D))$ attacks r' , contradicting $r' \in D'$.

Stable semantics. Let $\sigma = \text{stb}$. Let (P, D) be stable in T and \mathcal{E} as above. By Proposition 3 (P, D) is admissible in T , and by the proof above \mathcal{E} is admissible in F . It remains to show that \mathcal{E} attacks each argument $A \notin \mathcal{E}$ in T . Suppose the contrary, i.e., $A \notin \mathcal{E}$ is not attacked by \mathcal{E} . As \mathcal{E} contains all arguments based on (P, D) , (i) A is not based on (P, D) and (ii) no argument based on

(P, D) attacks A , and, in turn, (P, D) does not attack any $x \in \text{Prem}_d(A) \cup \text{DefRules}(A)$. If $\text{Prem}_d(A) \not\subseteq P$, we arrive at a contradiction: (P, D) would neither attack nor contain at least one ordinary premise in $\text{Prem}_d(A)$. If $\text{DefRules}(A) \not\subseteq D$, then there is a rule $r \in \text{DefRules}(A) \setminus D$. By definition of stability of (P, D) , it must be that r is not applicable by $(P', D') = (\mathcal{K}_p \setminus \text{att}(P, D), \mathcal{R}_d \setminus \text{att}(P, D))$. Since no defeasible element in A is attacked by (P, D) , we have $\text{Prem}_d(A) \subseteq P'$ and $\text{DefRules}(A) \subseteq D'$, implying that r is, in fact, applicable by (P', D') , a contradiction.

For the other direction, assume that \mathcal{E} is stable in F and (P, D) is not stable in T . Since \mathcal{E} is stable, \mathcal{E} is complete in F , and thus (P, D) is conflict-free in T . Since (P, D) is conflict-free and not stable, either (i) there is a $p \in \mathcal{K}_p$ neither in nor attacked by (P, D) or (ii) there is a rule $r \in \mathcal{R}_d$ s.t. r not in D , r not attacked by (P, D) , and r is applicable by $(P', D') = (\mathcal{K}_p \setminus \text{att}(P, D), \mathcal{R}_d \setminus \text{att}(P, D))$. In case (i) there is an argument $A = p$ which is not in \mathcal{E} , implying that \mathcal{E} attacks A , and hence (P, D) attacks p . In case (ii) there is an argument A with $r \in \text{DefRules}(A)$ and $\text{Prem}_d(A) \subseteq P'$ and $\text{DefRules}(A) \subseteq D'$. We have \mathcal{E} attacks A , contradicting the assumptions that A is composed of unattacked ordinary premises, defeasible rules, and the unattacked r .

Preferred semantics. Let $\sigma = \text{prf}$. Let (P, D) be preferred in T and \mathcal{E} as above. Suppose \mathcal{E} is not preferred in F . Then there is a preferred \mathcal{E}' in F with $\mathcal{E} \subset \mathcal{E}'$ (since \mathcal{E} is admissible in F). Further, there must be an argument $A \in \mathcal{E}' \setminus \mathcal{E}$ s.t. A is not based on (P, D) , since \mathcal{E} contains all arguments based on (P, D) . Thus there is an $x \in \text{Prem}_d(A) \cup \text{DefRules}(A)$ with $x \notin P \cup D$. By the above, $(P', D') = (\text{Prem}_d(\mathcal{E}'), \text{DefRules}(\mathcal{E}'))$ is admissible in T . Also, since each argument in \mathcal{E} is in \mathcal{E}' , we have $P \subseteq P'$ and $D \subseteq D'$, and $X \subset X'$ for one of $X \in \{P, D\}$. For this, consider any $y \in P \cup D$. Either $y \in P$ or $y \in D$ is applicable by (P, D) (by definition). Thus, there is an argument $B \in \mathcal{E}$ based on (P, D) with $y \in \text{Prem}_d(B) \cup \text{DefRules}(B)$ which is in \mathcal{E}' . Thus $(P, D) \sqsubset (P', D')$ and thereby (P, D) is not preferred in T .

For the other direction assume that \mathcal{E} is preferred in F and (P, D) is not preferred in T . Then there is a preferred assumption (P', D') with $(P, D) \sqsubset (P', D')$, since elements in T are finite: since (P, D) is admissible but not preferred, (P, D) is not \sqsubseteq -maximal, and one of the finitely many \sqsubseteq -greater assumptions must be admissible (otherwise (P, D) would be preferred). By iteration, there is a preferred assumption (P', D') with $(P, D) \sqsubset (P', D')$. Since (P', D') is preferred, we have $\mathcal{E}' = \{A \mid A \text{ based on } (P', D')\}$ is preferred in F (see above). Since $(P, D) \sqsubset (P', D')$, it follows that $\mathcal{E} \subset \mathcal{E}'$. For this, consider any $A \in \mathcal{E}$. A is based on (P, D) and on (P', D') . Suppose $\mathcal{E} = \mathcal{E}'$, then each argument A based on (P', D') is also based on (P, D) . This implies $P = P'$ (otherwise for $p \in P' \setminus P$ one would have an argument based on (P', D') but not on (P, D)). Thus, $D \subset D'$. Suppose that there is an $r \in D' \setminus D$ s.t. r is applicable by (P', D') . Then there is an argument B based on (P', D') with r as its top rule, contradicting that B is also based on (P, D) . Thus there is $r \in D' \setminus D$ not applicable by (P', D') , contradicting that (P', D') is a preferred assumption. Thus $\mathcal{E} \subset \mathcal{E}'$, a contradiction. \square

Acknowledgments

This work has been financially supported in part by Academy of Finland (grants 322869, 328718), University of Helsinki Doctoral Programme in Computer Science DoCS, and by the Austrian Science Fund (FWF): P30168-N31. Computational resources were provided by Finnish Grid and Cloud Infrastructure (urn:nbn:fi:research-infras-2016072533).

References

- Amgoud, L., and Besnard, P. 2019. A formal characterization of the outcomes of rule-based argumentation systems. *Knowledge and Information Systems* 61(1):543–588.
- Atkinson, K.; Baroni, P.; Giacomin, M.; Hunter, A.; Prakken, H.; Reed, C.; Simari, G. R.; Thimm, M.; and Villata, S. 2017. Towards artificial argumentation. *AI Magazine* 38(3):25–36.
- Baroni, P.; Gabbay, D.; Giacomin, M.; and van der Torre, L., eds. 2018. *Handbook of Formal Argumentation*. College Publications.
- Besnard, P., and Hunter, A. 2008. *Elements of Argumentation*. MIT Press.
- Besnard, P., and Hunter, A. 2018. A review of argumentation based on deductive arguments. In Baroni, P.; Gabbay, D.; Giacomin, M.; and van der Torre, L., eds., *Handbook of Formal Argumentation*. College Publications. chapter 9, 437–484.
- Bondarenko, A.; Dung, P. M.; Kowalski, R. A.; and Toni, F. 1997. An abstract, argumentation-theoretic approach to default reasoning. *Artificial Intelligence* 93:63–101.
- Brewka, G.; Delgrande, J. P.; Romero, J.; and Schaub, T. 2015. asprin: Customizing answer set preferences without a headache. In Bonet, B., and Koenig, S., eds., *Proc. AAAI*, 1467–1474. AAAI Press.
- Cerutti, F.; Gaggl, S. A.; Thimm, M.; and Wallner, J. P. 2018. Foundations of implementations for formal argumentation. In Baroni, P.; Gabbay, D.; Giacomin, M.; and van der Torre, L., eds., *Handbook of Formal Argumentation*. College Publications. chapter 15, 688–767.
- Charwat, G.; Dvořák, W.; Gaggl, S. A.; Wallner, J. P.; and Woltran, S. 2015. Methods for solving reasoning problems in abstract argumentation – A survey. *Artificial Intelligence* 220:28–63.
- Cyras, K.; Fan, X.; Schulz, C.; and Toni, F. 2018. Assumption-based argumentation: Disputes, explanations, preferences. In Baroni, P.; Gabbay, D.; Giacomin, M.; and van der Torre, L., eds., *Handbook of Formal Argumentation*. College Publications. chapter 7, 365–408.
- Dimopoulos, Y.; Nebel, B.; and Toni, F. 2002. On the computational complexity of assumption-based argumentation for default reasoning. *Artificial Intelligence* 141(1/2):57–78.
- Dung, P. M. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence* 77(2):321–358.
- Dvořák, W., and Dunne, P. E. 2018. Computational problems in formal argumentation and their complexity. In Baroni, P.; Gabbay, D.; Giacomin, M.; and van der Torre, L., eds., *Handbook of Formal Argumentation*. College Publications. chapter 13, 631–688.
- Gaggl, S. A.; Linsbichler, T.; Maratea, M.; and Woltran, S. 2020. Design and results of the Second International Competition on Computational Models of Argumentation. *Artificial Intelligence* 279.
- García, A. J., and Simari, G. R. 2004. Defeasible logic programming: An argumentative approach. *Theory and Practice of Logic Programming* 4(1-2):95–138.
- García, A. J., and Simari, G. R. 2018. Argumentation based on logic programming. In Baroni, P.; Gabbay, D.; Giacomin, M.; and van der Torre, L., eds., *Handbook of Formal Argumentation*. College Publications. chapter 8, 409–435.
- Gebser, M.; Kaminski, R.; Kaufmann, B.; Lühne, P.; Obermeier, P.; Ostrowski, M.; Romero, J.; Schaub, T.; Schellhorn, S.; and Wanko, P. 2018. The Potsdam answer set solving collection 5.0. *KI* 32(2-3):181–182.
- Gelfond, M., and Lifschitz, V. 1988. The stable model semantics for logic programming. In *Proc. ICLP/SLP*, 1070–1080. MIT Press.
- Heyninck, J., and Straßer, C. 2016. Relations between assumption-based approaches in nonmonotonic logic and formal argumentation. In *Proc. NMR 2016*, 65–75.
- Lehtonen, T.; Wallner, J. P.; and Järvisalo, M. 2019. Reasoning over assumption-based argumentation frameworks via direct answer set programming encodings. In *Proc. AAAI*, 2938–2945. AAAI Press.
- Maher, M. J. 2017. Relating concrete defeasible reasoning formalisms and abstract argumentation. *Fundamenta Informaticae* 155(3):233–260.
- Modgil, S., and Prakken, H. 2013. A general account of argumentation with preferences. *Artificial Intelligence* 195:361–397.
- Modgil, S., and Prakken, H. 2018. Abstract rule-based argumentation. In Baroni, P.; Gabbay, D.; Giacomin, M.; and van der Torre, L., eds., *Handbook of Formal Argumentation*. College Publications. chapter 6, 287–364.
- Niemelä, I. 1999. Logic programs with stable model semantics as a constraint programming paradigm. *Annals of Mathematics and Artificial Intelligence* 25(3-4):241–273.
- Prakken, H.; Wyner, A. Z.; Bench-Capon, T. J. M.; and Atkinson, K. 2015. A formalization of argumentation schemes for legal case-based reasoning in ASPIC+. *Journal of Logic and Computation* 25(5):1141–1166.
- Prakken, H. 2010. An abstract framework for argumentation with structured arguments. *Argument & Computation* 1(2):93–124.
- Prakken, H. 2012. Reconstructing Popov v. Hayashi in a framework for argumentation with structured arguments and Dungean semantics. *Artificial Intelligence and Law* 20(1):57–82.
- Snaith, M., and Reed, C. 2012. TOAST: online ASPIC⁺ implementation. In *Proc. COMMA 2012*, volume 245 of

Frontiers in Artificial Intelligence and Applications, 509–510. IOS Press.

Thimm, M., and Villata, S. 2017. The First International Competition on Computational Models of Argumentation: Results and analysis. *Artificial Intelligence* 252:267–294.

Toniolo, A.; Norman, T. J.; Etuk, A.; Cerutti, F.; Ouyang, W. R.; Srivastava, M. B.; Oren, N.; Dropps, T.; Allen, J. A.; and Sullivan, P. 2015. Supporting reasoning with different types of evidence in intelligence analysis. In *Proc. AAMAS 2015*, 781–789. ACM.

Visser, W. 2008. Implementation of argument-based practical reasoning. Master’s thesis, Utrecht University.

Yun, B., and Croitoru, M. 2016. An argumentation workflow for reasoning in ontology based data access. In *Proc. COMMA 2016*, volume 287 of *Frontiers in Artificial Intelligence and Applications*, 61–68. IOS Press.